Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Safe navigation with industrial AMRs among humans

Author
Marcell Balogh

Consultant
Dr. Attila Vidács

BUDAPEST, 2021

Table of contents

1. Introduction

Industrial robots took a long way to reach their current form and the new era of collaborative robots is currently rising. On the contrary of what people believed with the appearance of industrial robots, workers are still essential elements in the factories thus a stronger human-robot cooperation is started to emerge. Robots are able to cooperate even better with humans, taking their presence into account and cooperate with caution.

My goal is to extend the abilities of an autonomous mobile robot (AMR) for manoeuvring among people by taking into account their projected movement. With this safety feature AMRs could distinguish between people and static or other dynamic objects. It is expected to navigate in narrow corridors among human workers without impending production.

In this work I present a novel approach for sensor fusion where LiDAR and stereo camera information are combined in order to create a robust solution for human detection and avoidance. The implementation is based on the Robot Operating System (ROS) which gives the framework for modular development. The used sensors are an Intel RealSense depth camera, and an ydlidar laserscanner. These sensors are mounted on a custom built AMR platform.

In order to get accurate measurements, during development I test under real industrial conditions including narrow corridors, crowded workstations and dynamically changing lighting conditions. I compare and evaluate the sensory information both separately and with fusion, and also present its capabilities in real world.

2. Technological background

In this section I give an overview of the applied methods, and introduce the applied hardware elements. First, I take a look at existing technologies, how sensor fusion is implemented by both researchers and leading industrial companies, and show its typical use cases.

2.1 Related work

Combining multiple sensors are widely used in industrial robotics for various tasks and processes such as inspection and quality control, robot guidance, safety of workers, assembly lines, etc. Sensor fusion brings the input from each of different sensor types together, using software algorithms to provide the most comprehensive mapping from an environment, which is not possible to get from a single sensor. Experimenting with more fusion technics is an active field of current research topics.

A typical combination is working with separate colour and depth image stream. This can be provided by a single RGBD camera, which contain colour features along with depth information. Jafari et al faced a similar problem with detecting and tracking people real time from RGBD camera, from a moving observer [1]. Their article also takes into account the computational cost, in order to be useful for mobile robotic applications. They take advantage from depth information, using it to extract region-of-inerest, and extrapolate scene geometry.

Another regularly used technique is a combination of RGBD camera and a laserscanner. Meanwhile the camera is able to produce a 3D point cloud in a limited field of view (FOV), laserscanner can give a planar point cloud in 360 degrees. Cholakkal et al introduced a pipeline for more accurate depth estimation [2]. It projects lidar points into the camera frame, upsample the projection image with the help of depth map and applies a filter. Although their method is accurate, due to its cost effectiveness and GPU processing, on mobile robots this method is less applicable.

Among big companies, Amazon smart warehouses are famous for their automatization level, but still employs human workers for several tasks. To mention a fully automated warehouse, Ocado Smart Platform orchestrates swarms of bots in a giant storage grids with a clearance of just five millimetres between them [3]. It is also worth to mention that the smart parking systems, which store cars without human operators, providing an eco-friendly solution [4].

2.2 ROS

Robot Operating System (ROS) [5] is an open-source set of libraries and tools aiming to simplify the creation of complex robotic systems. It has a flexible approach to modular robotic architecture and provides APIs for many popular programming languages with proper support of transitions. The communication scheme follows distributed network communication patterns like the publish-subscribe or the request-response model, and there is a master node with dedicated role of orchestrating connections among nodes. A ROS environment consists of several different nodes that are responsible for separated task execution controlled by a master node.

Nowadays leading companies realized the strength of the open-source community and they also started to contribute and provide ROS APIs for most of their robotic devices and accessories. This led ROS to be the de facto standard of distributed robotic application development toolkit.

2.3 Lidar

Lidar stands for the acronym “light detection and ranging”, which is a sensing method using the reflection of pulsed laser to measure distances. It can generate precise two or three dimensional surface characteristics. For a 2D lidar, only one laser beam is required. With a spin movement, it can collect data on X and Y axes. These are suitable for performing accurate detection and ranging tasks, required by most of AMRs.

2.4 RGBD camera

RGBD cameras provide depth information besides the RGB colour stream. While 2D lidars detect objects in a horizontal plane, depth cameras have more versatile functionality. Creating a point cloud from depth camera stream could cover the surroundings not only in planar but in a wider range of observation.

A typical usage is to project point cloud to 2D plane to be able to handle these data as it would have come from a laser scanner. Another example for using colour and depth stream is to use colour stream for object detection and the depth information remain for distance measurement.

2.5 Sensor fusion

Perception modules are constantly getting information about their changing environment. These are heavily used in autonomous robotic applications like self driving cars or industrial robotics. During real-time environment monitoring, it is a common method to use the combination of perceptive sensors like lidars, monocular or stereo cameras, ultrasonic sensors.

In terms of sensor data fusion, different levels of abstraction appears during the implementation process. Low level fusion means fusing the raw sensory information. Combining a stereo camera and a lidar could mean a common point cloud from both devices. Applying this method for object detection would result in projecting lidar points into the camera image, associating points with the pixels [6].

Mid-level fusion used for object detection with independently captured sensor data. When both the lidar and the camera detect an obstacle, these positions will be fused in order to get accurate information about the environment. A typical approach is to implement a Kalman filter to alternate between prediction and update over and over again [7].

The high level fusion relies on the processed information from different sensors, where only the results (e.g. trajectory, detection) are fused. The downside of this process is that significant information loss could occur, thus it needs an application specific interpretation with caution.

2.6 Autonomous Mobile Robots

Automated Guided Vehicle (AGV) systems were developed into the key component of organized modern intralogistics since the 1950s. AGVs were initially used by the automotive industry in America. With the rapid development of sensory and regulatory technology and microelectronics paved the way for AGVs [8].

The main roles of these mobile robots are repetitive commissioning while carrying objects from A to B. They lack of on-board intelligence and obey simple programming instructions. In the past few years, a new kind of internal logistic system is started to take over from AGVs to Autonomous Mobile Robots (AMRs).

An AMR is equipped with a large variety of sensors and a powerful on-board computer to help understand its operating environment. Unlike the

AGVs which are limited to fixed route following along embedded wires or magnets, AMRs can adapt to environmental changes with processing sensor information real-time. They can safely perform the given task even in busy environment.

3. System design

This chapter introduces my method of sensor fusion, the steps of the high level design, and later the components are examined. It shows, how I formed different sensor inputs to fulfill my concept, and how I intended to test and demonstrate its usability and accuracy.

3.1 System architecture

ROS based systems fit well into the concepts of distributed robotic architecture. Ordering the processing tasks into different nodes gives a clear structure with transparent elements. Figure 3.1. introduces the high level architecture of how a sensor fusion worked out in the current setup, and shows the steps of applying mid-level fusion during realization.

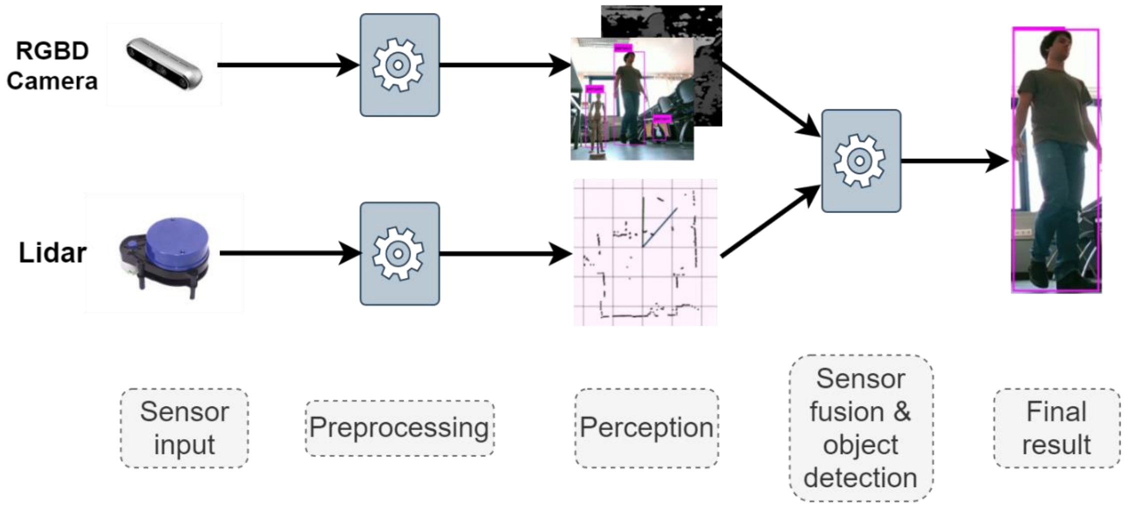


Figure 3.1. – Sensor fusion graph of camera and lidar

From the camera, I extracted both colour and depth stream simultaneously. On the colour stream I used a pretrained YOLO network [9] which detects and tracks people. The depth stream is then used for distance measurements. From the centre of the region of interest (ROI) coordinates, I calculate the spatial distance from the camera which later can be used in 3D location adjustments.

Another used sensor is a lidar, from the information is processed to detect human legs. It uses both legs to track instead of individual legs separately. I use a modified version of the available ROS leg detector package [10], which returns an angle of view where the feet can be seen.

After getting all the required data in the requested form from the camera and the lidar, the sensor fusion takes part. The idea is to validate the YOLO detector with the leg detector, in order to eliminate false positive detections, and increase the confidentiality level. With this metod, computational effort could be also reduced, because only those image regions are evaluated that are most likely to contain people.

3.2 Preprocessing sensor data

To implement a mid level sensor fusion, at first, sensory information needs to be preprocessed. The expected data from the camera is a colour and a depth image. An important aspect of camera images is that both stream needs to be aligned to each other. Because of the structure of stereo cameras, there are two lens responsible for depth recognition, and another separate lens is for colour streaming. These frames needs to be aligned in the beginning.

For synchronization and delay measurement between frames and other sensor inputs, timestamping is also required. The resolution of the image frames is also an important factor. Large network transport could exceed the actual bandwith capacity that results in delays and other unwanted effects. For prevention, several compression methods are available.

Laserscan points from the lidar comes unmodified. These are lightweight messages, not reaching average network bandwith limit. Timestamping here also necessary in order to be synchronized with other sensory information.

3.3 Navigation integration

To help the implementation of a robot navigation, ROS provides a metapackage called Navigation Stack [11] containing all the main elements required for a generic robot navigation in various environment. The components can be tailor made to work with various sensors and actuators like lidars, cameras, gps sensors, etc.

To be able to integrate detected people into the ROS navigation stack, a predefined `people_msgs/People` [12] message needs to be filled up with the appropriate parameters. After examination of the people message type, it contains the following fields: name, position, velocity, reliability, tagnames, tags. All the values can be computed from the sensor fusion.

Generally, right-hand rule could be a convential approach of AMR navigation. This would provide easy implementation by increasing the cost of moving left hand side, thus forcing robots to keep themselves in the right lane. In terms of AMRs, navigating in an industrial environment is not that simple. Workers take place everywhere in the corridors where machines need to be handled. Most of the cases, occupancy of corridors does not follow a pattern to AMRs scheduled and routed by. These circumstances necessitate AMRs to confidently yet safely handle these situations and navigate among humans.

With integrating the available sensory information into the navigation stack, mobile robots could be able to detect and track human workers and see wether they interfere with its planned trajectory. When required, a possible rerouting should be applied with respect to the extrapolated trajectory of the workers.

3.4 Demonstration scenario

The goal of my system setup is to present a robust sensor fusion for person detection and aviodance, that could provide easy integration into a robotic navigation stack. To achieve my objectives, a stereo camera and a laserscanner were mounted to an existing AMR platform.

To present its strengths and weaknesses, I show how it can detect and track people in different situations. The main goal is to combine the strength of both detection method, thus providing the current position of the people in sight for further actions.

My idea is to create a measurement area where the sensor accuracy could be recorded and compared each other. It makes possible to make modifications based on measurements.

4. Implementation

In this chapter I introduce how the previously mentioned methods took form through real word scenarios. The sensors take their final position, and ready to provide sensory information. I describe the processing methods both for stereo camera and for the lidar input, then the fuzed realization is introduced with example illustrations.

4.1 AMR platform

The previously introduced sensors are mounted to a custom built mobile robot, which is running on 4 holonomic wheels, each driven by separate servo motors. The camera is mounted to the front bumper of the AMR, meanwhile lidar takes place on the top of the vehicle where it can freely look around in 360 degrees. In order to get the best FOV from the camera, it is adjusted in such angle that it detect less floor, and give place for the upper content.

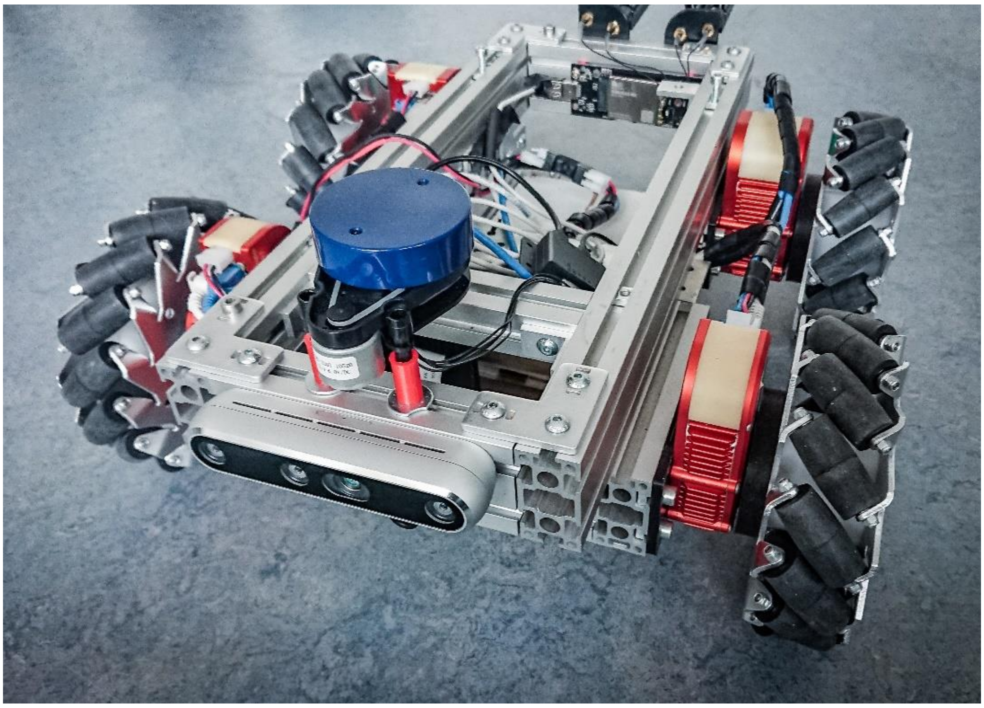


Figure 4.1. – Custom built AMR

The different coordinate systems of the sensors are adjusted in a universal robot description file (URDF) which is a conventional ROS robot representation. Here I defined the exact coordinate transformations of the camera and lidar related to the basis of the AMR. These transformations are an integral part of the

ROS ecosystem, it faciliates the handling of different sensor location, and the integration into the navigation stack.

In my case, these are only static transformations, because its position is fixed to another coordinate frame. The ROS tf package [13] also allows dynamic transformations which are useful at robots with dynamically moveable elements like robotic joints or end effectors.

4.2 Camera input

The mounted camera is an Intel RealSense D455 depth camera, which supports colour, depth and infrared image streams. To this camera Intel provides a RealSense SDK 2.0 [14] which is an extensive development kit supporting ROS wrappers, OpenCV, Point Cloud Library (PCL) etc. In my case the up to date and open source ROS wrapper is an important advantage from the aspect of integration.

After configuring the ROS environment, I compiled the camera driver and the ROS packages thus I could reach the various image sources from ROS framework. The input stream coming from the Realsense camera went through a compression. Instead of sensing the raw image data, JPEG compression is applied in order to prevent network bandwidth from overloading. The lossy compression results in significantly lower data while the content will not be deteriorated to negatively effect the object detection. This stream is then extracted and converted to an OpenCV image format, and then passed to the YOLO detector.

4.3 Object detection with camera

To detect and track humans from a real-time colour stream, I used Darknet, which is a high performance open source deep neural network framework written in C [15]. Besides other popular frameworks like PyTorch or Tensorflow, Darknet seemed ideal in this project, because it satisfied several constraints of ROS Melodic installation and it also include You Only Look Once (YOLO) implementation for real-time object detection and tracking [9].

YOLO is one of the most powerful method of real-time object detection with integration of advanced deep learning. It makes use of convolutional neural networks for the prediction of objects by using advanced mathematical formulations of image processing. I used and customized a pre-trained model on COCO and VOC dataset [16].

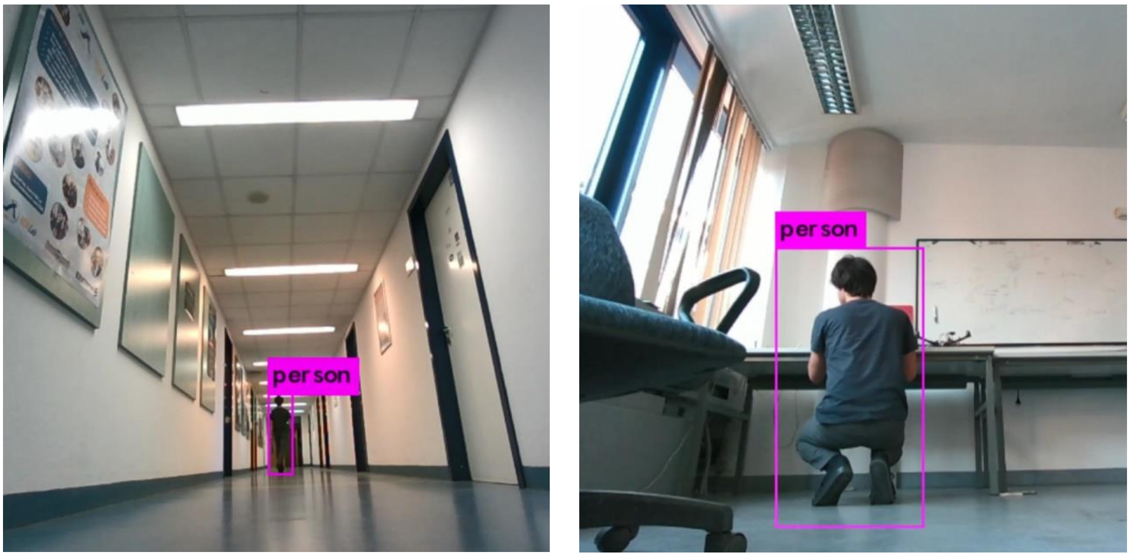


Figure 4.2. – Detection testing with camera in general environment

In order to demonstrate the abilities of the pretrained neural network, I tested it first in a general environment where I performed casual tasks not facing to the camera. In these situation I tried to act like people at factories whom the AMR could meet with, replacing items from A to B, working in different poses rearward facing to the camera, etc. but as Figure 4.2 shows, no matter how hard I tried, it does not effect the detection significantly.

To mention the downsides of these datasets, after cherry picking images from the training set, I met with images containing tagged persons from TV or other picture-in-picture scenes. On the left image of Figure 4.3, I could also face with false positive results. I placed a wooden mock-up model which has some 3D expansion, and I also placed a screen showing a person. Athough this can be very useful in various situations, according to my use case, these are unwanted tags that will have to be filtered out with sensor fusion.

At the right side of Figure 4.3, the depth image is presented. This information serves as the basis of distance measurement, and it makes possible to place the detected objects in a 3D map thus integrating obsctacles into the AMR navigation. After getting the coordinates of the purple bounding boxes, the distance value will be extracted from the centre of the ROI areas. In this case, without sensor fusion, it would contain two false coordinate with the mock-up and the screen coordinates.

Figure 4.3. – Person detection based on colour image and the corresponding depth image

4.4 Object detection with lidar

To process laserscan data from the lidar, I relied on the introduced method by Leigh et al [17], where a person detector is implemented based on a planar laser information. The presented solution is an extended modification of the official ROS leg detector package [10]. It uses a machine-learning-trained classifier to detect groups of laser readings as possible legs.

In my case, an ydlidar x4 2D lidar was available as presented on the top of the AMR in Figure 4.1, which has a detection range of 10 metres. Following the implementation, I provided the required topics for lidar data input, and the transformation coordinates. Then, I could extract the laserscan information, and visualized it in a ROS utility called rviz [18]. Rviz is a customizable 3D viewer where several imput sources can be visualized simultaneously.

The left part of Figure 4.4 shows the processed lidar data. The black dots are represented by the laser ray hitting and get reflected on an obstacle. As a result, the contours of the room become cleary visible.

The two straight lines which are forming an angle mark the leg positions of the detected person. This angle will limitate the detection area during fusion in regard to the colour image. The silhouette of the sceen is also visible by the small straight line on the top right part of the image. I attached the color frame next to the lidar representation for comparison purposes.

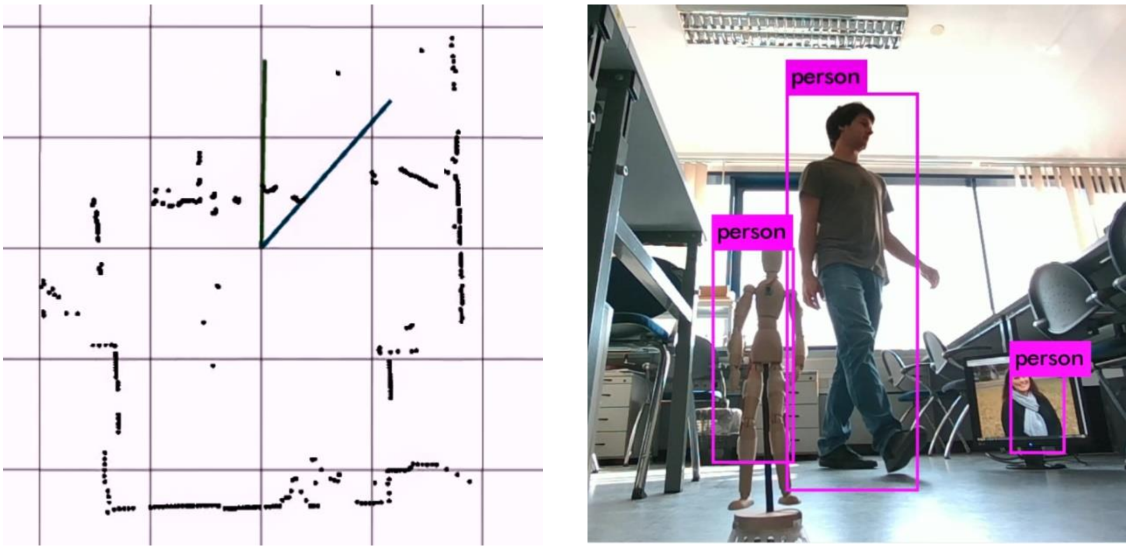


Figure 4.4. – Person detection based on lidar and the corresponding colour image

4.5 Sensor fusion realization

During my work I used a 2D lidar and a stereo depth camera to get information from the environment. Although the lidar has a 360 degrees field of view and the camera has only 87° vertical FOV, it is still an appropriate angle for my use case. As I experienced, this range is optimal for corridors, where mobile robots could encounter with people and forced to maneuvre.

During the first cases of fusion testing, several problems occurred that have to be solved. The most important is the time synchronization. Without time synchronization, separately recognised poses from the camera and the lidar could alter in time, wich would end up in inaccurate fusion. To avoid errors coming from mismatched timestamps, I regularly compare the header information of the lidar and camera information. The conventional ROS message types are in case of the camera is `sensor_msgs/Image`, and the lidar message type is `sensor_msgs/LaserScan` [19]. According to their description, both starts with a header, containing this timestamp.

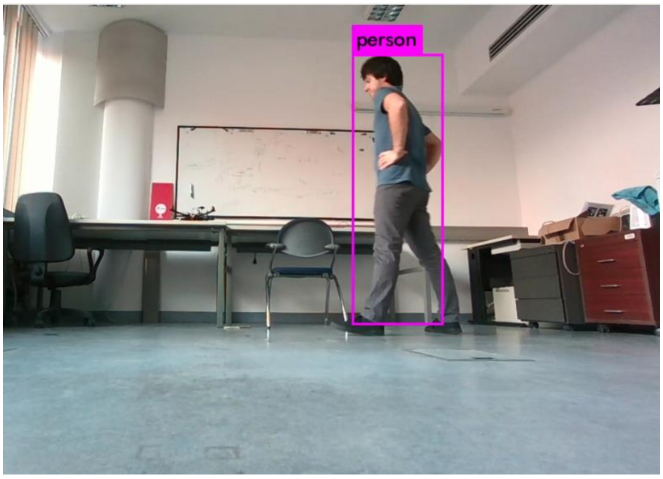
Another issue was the stream size coming from the camera node. By default, ROS operates with uncompressed image stream which can easily overload the network bandwidth. For comparison, I created and summarized bandwith measurements in Table 1, showing the applied compression and quality parameters.

Source	Quality	Bandwidth
raw colour image	bgr8	222,24 Mbps
raw depth image	mono16	110,48 Mbps
compressed colour image	100% JPEG	32,32 Mbps
compressed colour image	15% JPEG	1,686 Mbps
compressed depth image	100% JPEG	11,84 Mbps
compressed depth image	15% JPEG	2,074 Mbps

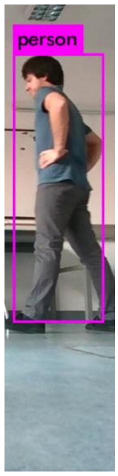
Table 1.– Measured transmission parameters

I find a 15% JPEG quality a good trade-off between image size and quality. As a result, transferring a compressed JPEG frame results in such a small size that could be transferred without difficulties. Therefore I applied JPEG compression with 15% quality for both colour and depth stream. With these modifications, timestamps can be matched without significant delay.

After facing all the issues, the fusion worked correctly, as Figure 4.5 illustrates. The before image have been cut as a result of the leg detector, and YOLO detector run only on the remained snippet. Important to mention that cutting the image only modifies the width of the image, height remain unchanged.



a)



b)

Figure 4.5. – Example result of the sensor fusion with before a) and after b) comparison

5. Experimental results

This chapter contains the achieved results and gives an insight to the measurements. First, the measurement validation method is explained, how I tested the sensor fusion. After collecting enough information, I summarize and evaluate the retrieved data, and I share my experiences.

5.1 Measurement arrangement

To get more accurate measurement results I set up a test environment with annotations that can be recognised during lidar and depth measurements. The test setup was a 2m long line where people can walk, as on Figure 5.1 shows. The distance of the mounted sensors of the mobile robot are also accurately arranged to the scenario.

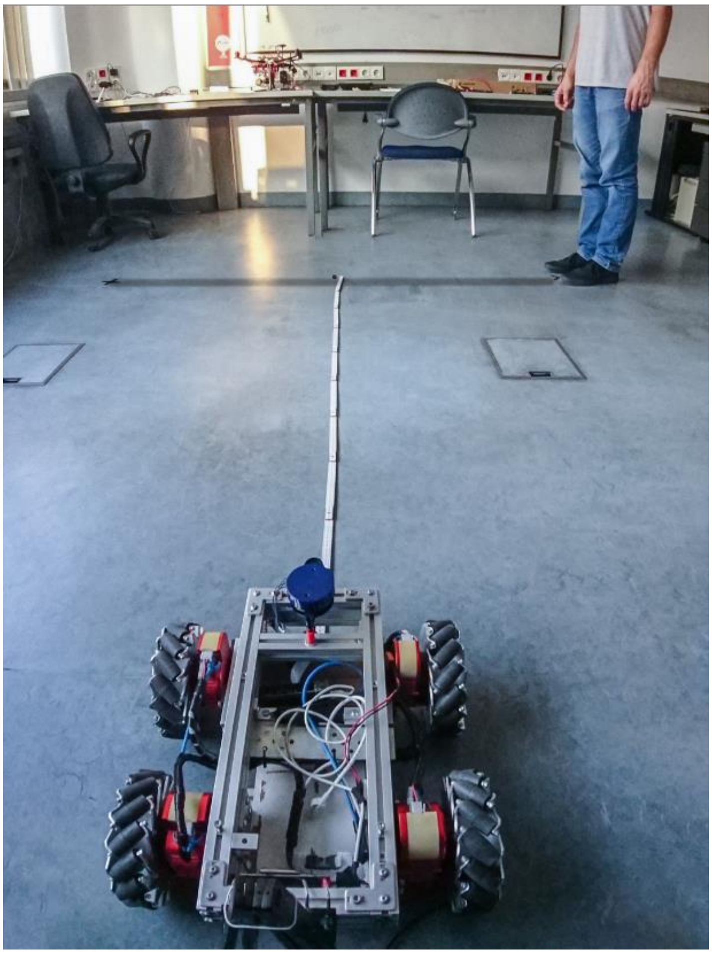


Figure 5.1. – Measurement setup with annotations

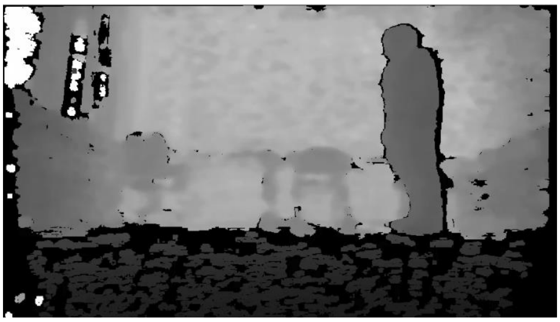
My goal with this scenario is to collect relevant data about scenarios that are easy to understand and evaluate. Thus I can show separately the detection accuracy of the yolo and the leg detector, then compare their results. Based on these charts, weighting can be modified for the mutual benefit of the fusion. When a detector seems to be more robust and accurate, it can be counted with greater weigh than the other.

5.2 Results and evaluation

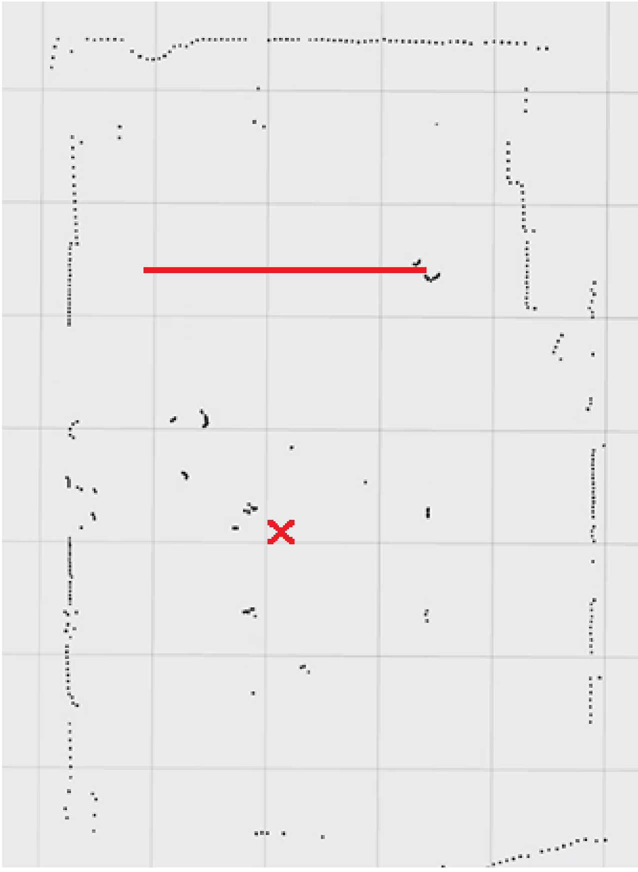
To accomplish validation I used the introduced test scenario to check how the detector could follow a person while walking. This challenges both person and leg detectors.

I created a script that matches timestamps and extract x and y coordinates from both detectors. I plot these data into a coordinate system to examine the accuracy of the separate detectors.

a)



b)



c)

Figure 5.2. – Measurement viewed from camera a), b) and from lidar c)

Figure 5.2 shows how my measurement setup looks like from a ROS environment, to measure linear walking detection accuracy. The red line on c) represents the expected trajectory of the straight walking. The red X represents the middle position of the sensors.

The results are collected to separate charts as Figure 5.3 shows. YOLO detector seems accurate for line following. At the starting point and at the end there are some inaccurate measurements, but it could be caused by the turnovers. The scattering of the x,y coordinates seems homogen, which provide reliable base of measurement.

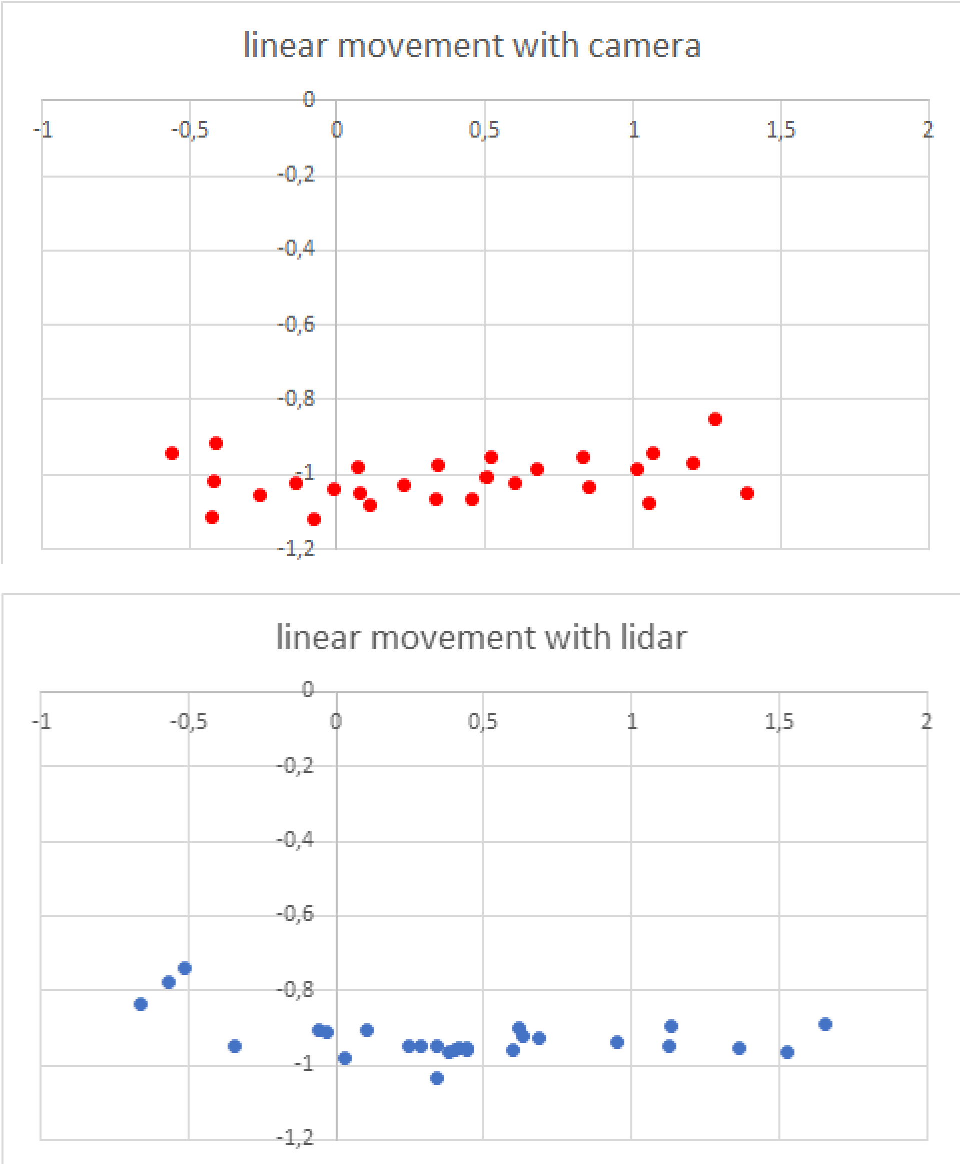


Figure 5.3. – Results of linear movement captured by camera (up) and lidar (down)

For experiment purposes I run detection during walking in a circle. At first sight it is clearly visible that the leg detection is more shaky than the person detector. This waving caused by its calculation method. The coordinate of the separate legs is calculated by averaging. Acording to this, walking in a straight line would also look like a wavy line.

Distance measurement from YOLO detector takes the median depth value of the ROI to calculate with, and extract the exact distance. This results in a much normalised trajectory as the orange line shows in Figure 5.4. Instead of a more round trajectory, these waves are caused by multiple factors. It consists of the uncertainty of walking around, the leg measurement, which could contain such inaccuracy at overlapping legs, and in case of YOLO detector, the loose outfit that does not follow the exact shapes of the human body. It also visible that yolo detection could contains a constant error.

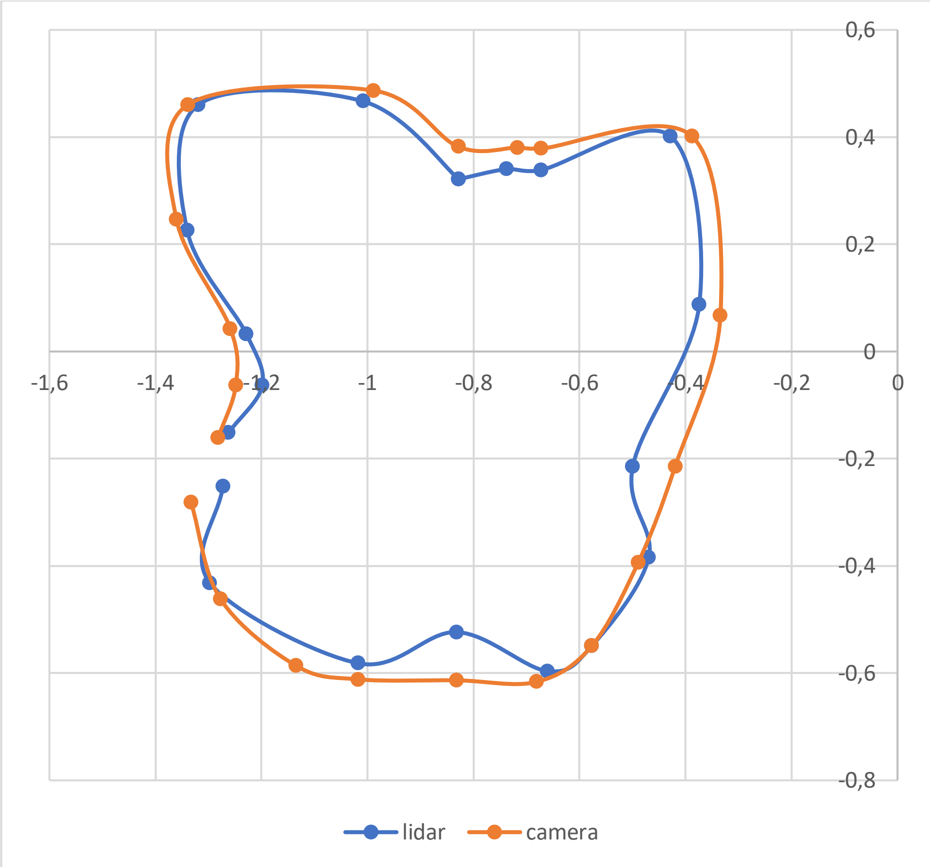


Figure 5.4. – Comparing detector trajectories

5.3 General experience

I was more than satisfied with the results during the on-site measurements. Contrary to my concerns, it turned out that wearing an ESD protective overall covering my full body combined with a face mask still provide enough feature for human detection. as the images on Figure 5.5 show. The leg detector had to follow overlapping legs, with sightly bigger diameter as usual. Both detectors could recognise people in the same way as they do in a regular environment.

However, the cleanroom was really clear and organised, the different structure of machines and equipments are able to form such a way that it could be interpreted as human feature. This is presented on image b) of Figure 5.5. This is also a good representation of relying on only the YOLO detector. Applying the fusion with lidar saved the scenario, and found the human legs correctly.

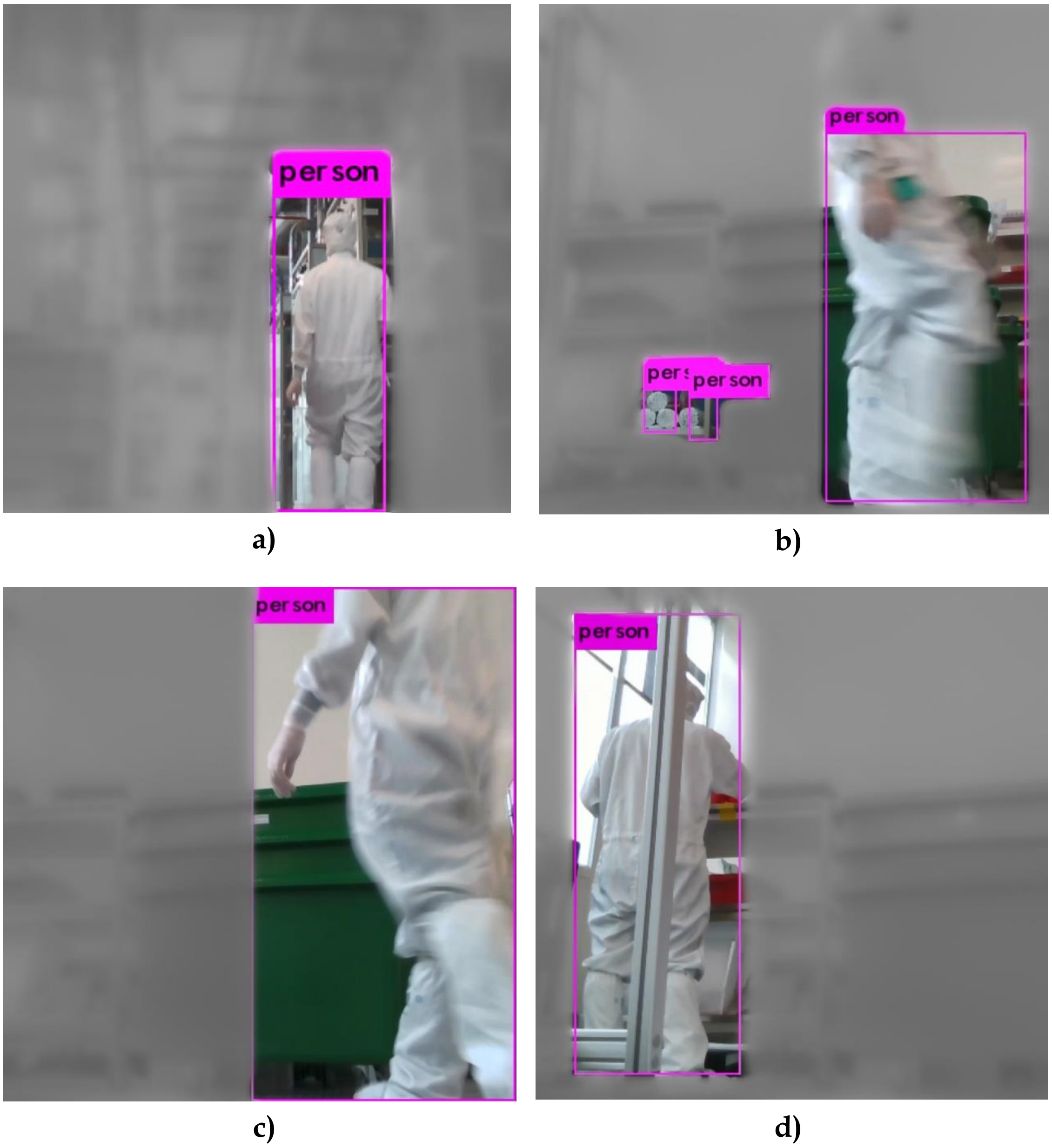


Figure 5.5. – Detection testing in industrial environment[1]

[1] Although the detection could be performed in industrial environment, I had to blur the background of these images because of company request.

6. Summary, future work

Throughout this project I introduced a sensor fusion method that combines the advantages of RGBD cameras and lidars to provide a safe yet robust solution for person detection in an industrial environment.

I created a mid-level sensor fusion architecture which contain a person detector based on colour camera stream, and a leg detector based on lidar measurements. After mounting the sensors to a custom built AMR, I integrated the processing nodes to be able to communicate each other. I successfuly faced the challenges of preprocessing and conditioning sensor input for the required format, and then extract the requested information, namely the coordinates of the detected objects in a common coordinate system. With all the calculated and matched coordinates, the social layer can be integrated to extend ROS navigation capabilities.

For future development, not just the current position but an extrapolated trajectory of a detected object or objects could help the trajectory planning and rerouting. The navigation also can be extended with locating the other working mobile robots to be able to calculate at trajectory planning.

Acknowledgement

I would like to thank the work of Dr. Attila Vidács, who helped me giving all technical support and advise I needed. I am thankful to my family for their sustained encouragement. Without them and my close friends this thesis would not have been possible. My thanks to everyone!

List of figures

Literature references

[1] O. H. Jafari, D. Mitzel and B. Leibe, "Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras," 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 5636-5643

[2] H. H. Cholakkal, S. Mentasti, M. Bersani, S. Arrigoni, M. Matteucci and F. Cheli, "LiDAR - Stereo Camera Fusion for Accurate Depth Estimation," 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), 2020, pp. 1-6,

[3] Pioneering the future through serial technology innovation, online, *https://www.ocadogroup.com/technology/technology-pioneers*, accessed: 26.10.2021.

[4] Fully Automated Car Parking Systems, *https://www.pouted.com/fully-automated-car-parking-systems*, online, accessed: 26.10.2021.

[5] Stanford Artificial Intelligence Laboratory et al. “Robotic Operating System”, *https://www.ros.org*, 23.05.2018.

[6] Types of Sensor Fusion Algorithms, online, *https://www.thinkautonomous.ai/blog/?p=9-types-of-sensor-fusion-algorithms*, accessed: 26.10.2021.

[7] Sensor Fusion - LiDARs & RADARs in Self-Driving Cars, online, *https://www.thinkautonomous.ai/blog/?p=sensor-fusion* , accessed: 26.10.2021.

[8] Günter Ullrich, Automated Guided Vehicle Systems: A Primer with Practical Applications, 2015

[9] Redmon, Joseph and Farhadi, Ali. “YOLOv3: An Incremental Improvement,” 2018

[10] ROS People detection package, leg detector, online,. *http://wiki.ros.org/leg_detector*, accessed: 26.10.2021.

[11] Eitan Marder-Eppstein and Eric Berger and Tully Foote and Brian Gerkey and Kurt Konolige, “The Office Marathon: Robust Navigation in an Indoor Office Environment,” 2010 IEEE International Conference on Robotics and Automation (ICRA), 2010

[12] ROS people stack documentation, people messages, online *http://wiki.ros.org/people_msgs* , accessed 27.10.2021.

[13] Tully Foote, “tf: The Transform Library”, 2013.

[14] Intel RealSense SDK 2.0, online, *https://www.intelrealsense.com/sdk-2*, accessed: 26.10.2021.

[15] Joseph Redmon, “Darknet: Open Source Neural Networks in C, “ *http://pjreddie.com/darknet*, 2013-2016

[16] M. Bjelonic "YOLO ROS: Real-Time Object Detection for ROS", *https://github.com/leggedrobotics/darknet_ros* , 2018

[17] A. Leigh, J. Pineau, N. Olmedo and H. Zhang, "Person tracking and following with 2D laser scanners," 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015

[18] ROS visualization documentation, rviz, online *http://wiki.ros.org/rviz* , accessed: 26.10.2021

[19] ROS standard message types, online *http://wiki.ros.org/sensor_msgs* , accessed: 26.10.2021